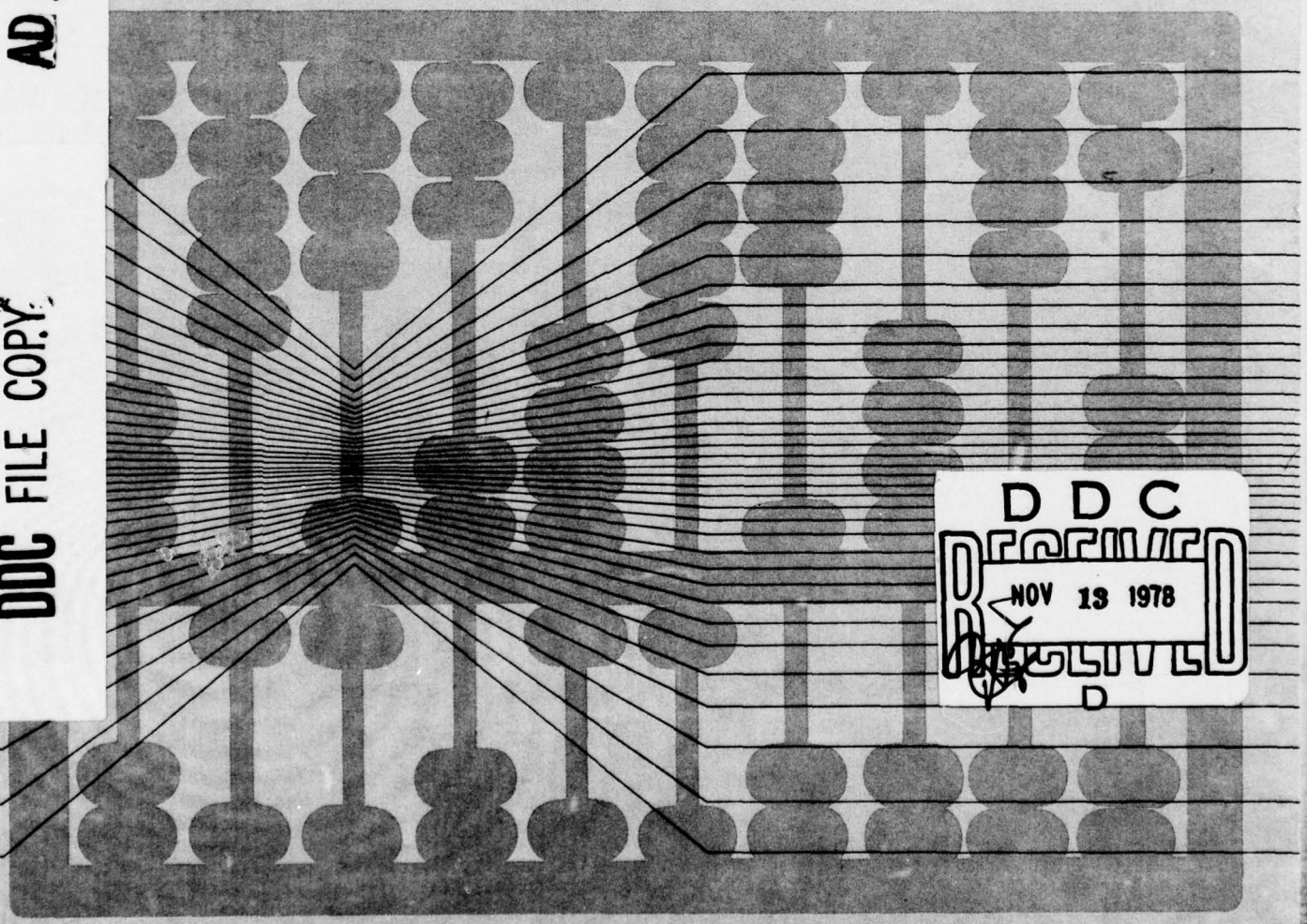
LEVEL II

I.P. Sharp Associates

AD A061052

DDC FILE COPY

EUCLID COMPILER
QUARTERLY REPORT No 2



LEVEL II

(9) Quarterly Technical Report. no. 2,

for

(6) EUCLID Compiler for PDP-11.

Number 2

PERIOD COVERED: 1 Apr [redacted] to 30 June 78

(11) 24 Jul 78

This research was sponsored by the Defense Advanced Research Projects Agency under ARPA Order [redacted]-3475
(15) Contract No. MDA-903-78-C-0037
Monitored by Steve Walker
Effective Date of Contract 1 Oct 1977
Contract Expiry Date 31 Mar 1979

A portion of this project is being sponsored by the Canadian Department of National Defence, Chief of Research and Development

(12) 28 p.

(14) IPSA-3819-912

(10) David Bonyun R. C. Holt
I.P. Sharp Associates Limited
Suite 600
265 Carling Avenue
OTTAWA, Canada K1S 2E1

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* EUCLID Compiler for PDP-11 Report Number 2	5. TYPE OF REPORT & PERIOD COVERED Quarterly Technical to 30 June 1978 5. PERFORMING ORG. REPORT NUMBER IPSA-3819-002
7. AUTHOR(s) D.A. Bonyun, I.P. Sharp Associates R.C. Holt, University of Toronto	8. CONTRACT OR GRANT NUMBER(s) MDA-903-78-C-0037
9. PERFORMING ORGANIZATION NAME AND ADDRESS I.P. Sharp Associates Limited 600-265 Carling Avenue OTTAWA, Ontario, Canada K1S 2E1	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Information Processing Techniques Office DARPA ARLINGTON, VA 22209	12. REPORT DATE 24 July 1978 13. NUMBER OF PAGES 26
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*	15. SECURITY CLASS. *(of this report)* UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

DISTRIBUTION STATEMENT A
Approved for public release; Distribution Unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

Part of this project is funded by the Canadian Department of National Defence

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

EUCLID, compiler, computer security

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The work towards a EUCLID compiler is progressing satisfactorily. Communication with Ford Aerospace (KSOS team) has resulted in very slight direction changes in the short run to satisfy their immediate needs.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

Quarterly Technical Report #2
EUCLID Compiler Project

Table of Contents

Quarterly Technical Report #2

EUCLID Compiler Project

Report Summary

This second quarterly technical report covers the period from 1 April 1978 to 30 June 1978. During that time three major steps were taken:

(a) the general designs developed earlier became much more detailed providing specifications for the individual passes of the compiler.

(b) implementation (coding) of the four remaining passes began and is well under way.

(c) communications with the principal users of the compiler (the KSOS team at Ford Aerospace) were firmly established so that their needs, in terms of both technical requirements and schedule, could be understood and met by the implementation team at the University of Toronto.

As a result of this communication the product known as Middle EUCLID which was to have been delivered at the end of July has been postponed. Instead, Ford Aerospace agreed to the delivery of an "October EUCLID" in October embodying their requirements. In the interim, as pieces of the compiler are fitted together to provide stand alone iterations towards October EUCLID, these will be delivered informally to Ford.

Quarterly Technical Report #2

EUCLID Compiler Project

I. BACKGROUND

The work described herein is part of the effort to achieve a compiler for the language EUCLID for the PDP-11. This work is part of the major project to achieve a secure operating system. Ford Aerospace has been given the contract to develop such an operating system for the PDP-11 to be known as KSOS (Kernalized Secure Operating System) and they are seen as the principal users of the EUCLID compiler.

A portion of the EUCLID project is being funded by the Canadian Department of National Defence (DND). The project-team is located in Canada: the work is essentially being done at the University of Toronto by people in the Computer Systems Research Group in conjunction with I.P. Sharp Associates Limited. The monitoring is being done jointly by DND and by ARPA.

II. THE APPROACH

The work of the past three months may be divided into a number of separate sections. These are:

(i) Design;

(ii) Implementation;

(iii) Communications with Ford;

and (iv) Communications with the EUCLID Committee.

We shall consider each of these separately and in turn.

(i) Design

From the general design of the compiler which has evolved and is to be found piecemeal in the working papers (See Appendix B), a more detailed set of specifications has been developed. These specifications are for the four passes of the compiler (known as the Builder, the Conformance Pass, the Allocator, and the Coder) which follow the two already well established (the Scanner and the Parser).

The Builder is concerned with creating the symbol table and type tables, and appropriately modifying the token stream which flows successively through the remaining passes until the Coder provides object code.

The Conformance Pass determines the semantic legitimacy of the source code and uses the type-sameness rule mentioned so often in the message file (Appendix C). This pass together with the Builder constitutes the semantic portion of the compiler.

The Emitter passes - the Allocator and the Coder - are the machine dependent portion of the compiler. Up to this point, although some reference must be made to the host machine, the compiler is largely machine independent.

(ii) Implementation

The actual implementation of these four remaining passes is well under way. It is when these passes have been coded and integrated with one another that serious testing and enhancement may take place. It is the general philosophy that an initial version will be put together which omits many of the features ultimately to be supported. Once this version is shown

to work satisfactorily, the features will be incrementally incorporated until the final version is finished. The first stand alone version is expected at the end of July or early in August, but may be later.

(iii) Communications with Ford

As the result of establishing a dialogue with the principal user of the compiler (the Ford Aerospace team writing KSOS) two significant items need reporting.

(a) The needs of the Ford team are much better understood now by the implementation team. As a result the order of the increments to the compiler has been slightly altered to conform to the expressed requirement of the team. Moreover, the version known as Middle EUCLID has been postponed and redefined as October EUCLID. This will now be delivered to Ford at the beginning of October (or before) and will contain an agreed minimal set of features. (Appendix C defines Middle EUCLID, while Appendix D defines the revised version.)

(b) The experience of the Toronto team in writing EUCLID code and in writing the compiler will be transferred to Ford by means of a tutorial to be held in early August at Palo Alto. This is to improve the Toronto team's sensitivity to the needs of the KSOS team, as well as to provide the KSOS team with the benefit of Toronto's acquired knowledge of the peculiarities of EUCLID.

(iv) Communication with the EUCLID Committee

Because the language is now much more stable, the level of communications between the committee and the Toronto team has diminished. Appendix E lists the ARPANET messages since the beginning of 1978. Although there has been a flurry recently, this has centered principally on the revised report which the committee has issued. It is expected that this revised report will be available to those requiring it from I.P. Sharp Associates. It will be made publically available when this draft revision has been generally corrected and accepted.

III. WORKING PAPERS

The current index of working papers is shown as Appendix B.

APPENDIX A

Progress Report No 2

(1 April - 30 June 1978)

During the period April 1, 1978 to June 30, 1978 the EUCLID implementation project was primarily concerned with developing detailed designs from previous general designs, and with the beginning of implementation of semantic passes. During this period, communication with the secure operating system project at Ford Aerospace produced a detailed list of their requirements for the EUCLID translator.

The next major goal is the production of a EUCLID translator that produces PDP-11 code, and which can translate (bootstrap) itself. The translator will consist of six major passes: the Scanner and Parser (already constructed and tested), two semantic passes (the Builder and Conformance), and two emitter passes (the Allocator and Coder).

The progress during this period included:

> Detailed design of the Builder pass produced revised specification of the disk-resident symbol and type tables, as well as detailed documentation of the Builder-produced token stream.
>
> Detailed design of the Conformance semantic pass produced specification of tests (e.g., arithmetic type checking) to be performed by the pass.
>
> The Builder and Conformance passes entered initial stages of implementation.
>
> Detailed design of emitter mechanisms produced: set operation code templates, routine call/return code templates, preliminary register allocation, Boolean

operation code templates, scheme for addressing modules and extended parameters, and scheme for compile-time representation of variable/constant values.

R.C. Holt
CSRG
13 July 1978

APPENDIX B

Working Paper Index

Number	Title
1	On Legality Assertions in EUCLID
2	A Possible EUCLID Compiler Structure
3	Structure of the Scanner and Screener
4	Programming Conventions
5	A Syntax/Semantics Language
6	Small EUCLID
7	The Syntax of Small EUCLID and Small C
8	Screener Output Files
9	File Input/Output Routines
10	A Child's Guide to Imports and Exports
11	A User-Oriented Syntax of Full EUCLID
12	A Discussion of "A User-Oriented Syntax of Full EUCLID
13	Format of the Syntax/Semantic Tables
14	A Run-Time Model for EUCLID
15	EUCLID Language-Defined Identifiers
16	Notes on EUCLID Compiler Structure
17	SUE/11 Procedure Linkage (SUE.8 Working Paper 9)
18	Constant Folding in Postfix Expressions
19	Syntax of the Parser Output for the Full EUCLID Translator
20	Input/Output in EUCLID
21	Some Major Tasks for Jan 1 - July 1, 1978 EUCLID Implementation
22	The Sizer - A Part of the Conformance Checker

Number	Title
23	The Dot Interpreter - A Part of the Builder
24	The Constant Folder
25	Evaluating Literals and Folding Constants
26	Proposed EUCLID Translator Structure
27	Notes on the Structure of the EUCLID Translator
28	Value Descriptors
29	Type Table - Detailed Description
30	Symbol Table - Detailed Description
31	Variant Records and Discriminating Cases
32	Interface to Disk-Resident Tables
33	Set Operations in Small EUCLID
34	Compiling Parameterized Types
35	A Symbol/Type Table Example Involving Variant Records
36	A Specification for the Type Sameness Enforcer
37	What "Well-Behaved Arithmetic" Means to the Compiler
38	Legality Specifiers
39	More About Compiling Parameterized Types: Pass Responsibilities and Design Details
40	Export of Types
41	Details of Parameterizers and Initializers for Variant Records
42	Source Inclusion Facility in Toronto EUCLID
43	Visibility Aspects of Access Control
44	EUCLID Set Operations
45	Addressing and Routine Calling in EUCLID/11
46	A Code Template for Case Statements
47	Addressing and Routine Calling in the C Language

Number	Title
48	A Preliminary Allocation of Registers for Toronto EUCLID/11
49	Translating Boolean Expressions into Control Flow
50	Symbol and Type Table Representation of Declared Symbols
51	Allocation and Addressing in the Emitter Passes
52	Symbol and Type Table Representation of Type Definitions
53	Features of Middle EUCLID
54	Middle EUCLID Subset
55	A Set of Exception Handling Routines for EUCLID
56	The Locator List (or Scope) Table
57	Revised Contents of the Symbol Table
58	Data Descriptors for Use by the Emitter
59	Tests to be Performed in Conformance Pass
60	Value Descriptors Revisited
61	A Proposed Design for the EUCLID Paragrapher
62	Token Stream Output of the Builder
63	Revised Contents of the Type Table
64	A Code Template for Indexed For Loops
65	Code Templates for Long Moves and Long Compares

APPENDIX C

Middle EUCLID

(Working Paper 54)

The continuing evolution of the EUCLID language has caused some delay in the original EUCLID implementation schedule. This working paper attempts to document the EUCLID language features that will not be supported by the prototype translator that will be delivered.

A language feature that is DEFERRED will definitely not be supported by the prototype translator. Language features that MAY BE DEFERRED will be supported by the prototype translator if the Implementation Team has sufficient time to implement them.

A meeting in January 1978 between the EUCLID committee and the EUCLID implementation team resulted in some substantial changes to the language. The prototype translator will implement the language as revised by this meeting. The EUCLID Committee under-took to produce a revised report on the EUCLID language by mid March 1978. It is now expected that this revised report will be available in late June 1978.

The EUCLID subset supported by the prototype translator will be somewhat larger than Small EUCLID and will be tailored to the needs of the Implementation Team and the EUCLID Users. The subset contains all the features of Small EUCLID, and is called Middle EUCLID.

The remainder of this paper lists the deferred and possibly deferred language features and also notes major changes that are a result of the January 1978 meeting. For ease of reference, the section numbers and headings of the February 1977 EUCLID report have been used.

3.2 Legality Assertions

Legality assertion generation is DEFERRED.

5. Manifest constants

The formal parameters of a type, when used in the body of the type, are never considered to be manifest even if the corresponding actual parameters for an instance of the type are manifest constants (JAN 1978).

6.1.1 Enumerated types

The functions T.Min and T.Max will be restricted to exactly two arguments. The Min and Max functions MAY BE DEFERRED.

6.1.2 Standard simple types

The concept of well-behaved and the arithmetic operations for unsigned integers have been revised and restricted (JAN 1978).

The spelling of standard type names has been revised so that all standard type names now begin with a capital letter (JAN 1978).

6.1.3 Subrange types

If an integer subrange type has non-manifest bounds then it must be contained in SignedInt when the bounds are evaluated.

6.2.1 Array types

The implementation of arrays with non-manifest bounds MAY BE DEFERRED.

If the componentType of an array is a subrange type then that subrange type must have manifest bounds (JAN 1978).

6.2.2 Record types and String Type

Subrange types used to declare variables and constants in a record definition must have manifest bounds (JAN 1978).

Arrays with non-manifest bounds inside records MAY BE DEFERRED.

Direct nesting of variants (i.e. case within a case) is DEFERRED. The same effect may be achieved by wrapping a record declaration around the inner case.

The itsTag standard component has been added to allow reference to the value of a variant records tag (JAN 1978).

The tag of a variant record definition must be exactly an identifier corresponding to a formal parameter of the record type. Manifest constant expressions used in the tag field are DEFERRED.

Labels in variant record cases are restricted to being literal constants, named literal constants, or sub-ranges thereof. Manifest constant expressions used as variant case labels are DEFERRED.

The standard type String is supported and is changed to be simply a packed array of characters with the upper bound of array being a parameter to the type (JAN 1978).

6.2.3 Module types

Validation of import and export restrictions will be DEFERRED.

A compiler-generated THUS list has been added to allow the compiler to supply the transitive completion of imports lists written by the programmer. Identifiers imported via THUS lists may not be directly used by the programmer. (JAN 1978). THUS list generation is DEFERRED.

Parameterized module types are DEFERRED.

Assignment and comparison of module variables are DEFERRED. Module variables as parameters to routines are DEFERRED. Collections and arrays of modules are DEFERRED.

Enforcement of type opaqueness MAY BE DEFERRED.

The FINALLY clause for modules MAY BE DEFERRED.

The use of the standard component ItsType on module variables is DEFERRED.

Execution of the invariant assertion for modules MAY BE DEFERRED.

The abstraction function declaration in a module definition is DEFERRED.

Parentheses are required around an expression following the keyword INVARIANT.

Enforcement of EUCLID prohibitions on the use of imported variables in the declaration part of modules is DEFERRED.

6.2.6 Pointer and Collection types

Use of zones other than the standard System Zone MAY BE DEFERRED.

Collections with non-manifest object types and uses of UNKNOWN to define the object type of a collection MAY BE DEFERRED.

Reference counted collections and CHECKABLE collections (JAN 1978) will be DEFERRED.

6.3 Parameterized types

The type of formal parameters to a parameterized type are restricted to being simple manifest types (JAN 1978).

Non-manifest actual parameters to types MAY BE DEFERRED. Any use of a type formal parameter other than as a tag of a variant record case MAY BE DEFERRED.

The use of FORWARD in type definitions MAY BE DEFERRED.

6.4 Type compatability

The type sameness rule has been substantially modified (JAN 1978).

6.5 Explicit type conversions

The type converter "<<=" has been deleted. A functional notation for type conversion has been added. (JAN 1978).

7. Constants and variables

Structured constants must be manifest (JAN 1978). Structured records and module constants MAY BE DEFERRED.

7.4 Binding

Enforcement of the non-overlap rule for binding MAY BE DEFERRED.

The bind constant MAY BE DEFERRED.

9.1.4 Assert statements

Parentheses are required around an expression following the keyword ASSERT.

9.2.3.2 For statement

Set and module generators are DEFERRED.

10. Procedures

Code blocks can contain only Unix assembler statements. The code in these blocks can refer to routine parameters but not to imported variables.

Routine formal parameter definitions that involve other routine parameters will be DEFERRED. PARAMETER is supported.

INLINE procedures are DEFERRED. Procedures declared using INLINE will be temporarily implemented out of line.

The use of FORWARD in routine definitions MAY BE DEFERRED.

Parentheses are required around an expression following the keywords PRE and POST.

The Toronto implementation allows the keyword EXTERNAL to be used in place of the keyword FORWARD to permit linking to separately compiled C and Assembler routines.

11. Function declarations

The resultName in the function header is required rather than optional (JAN 1978).

Functions returning non-scalar values with manifest types MAY BE DEFERRED.

Functions returning scalar values with non-manifest types MAY BE DEFERRED.

Functions returning non-scalar values with non-manifest types are DEFERRED.

Parentheses are required around the expression following the keyword RETURN.

INLINE functions are DEFERRED. Functions declared using INLINE will be implemented out of line.

Parentheses are required around an expression following the keywords PRE and POST.

The Toronto implementation allows the keyword EXTERNAL to be used in place of the keyword FORWARD to permit linking to separately compiled C and Assembler routines.

APPENDIX D

Revisions to Middle EUCLID

Summary of the 23 June 1978 telephone conference between the EUCLID Implementation Team (R.C. Holt, D.B. Wortman, and D.A. Bonyun) and the KSOS implementors (T. Berson, K. Biba, M. Pliner, R. Feiertag).

The language points raised in Berson's position paper were discussed and the following agreement was reached.

1. The implementation will try to implement functions that return values with non-scalar manifest types as soon as possible. Functions that return non-scalar non-manifest types will remain deferred.

2. The implementation will include (at least) "simple" parameterized module types.

3. Forward type declarations will be implemented.

4. "Simple" binds (e.g. binds to array elements or to records) will be implemented.

5. Set generators are easy to do but KSOS doesn't need them very much. Module generators are a lot harder and KSOS needs them to encapsulate its data structure handling. The implementation team feels that there may be serious implementation problems with module generators. They are unwilling to proceed with a module generator implementation until these problems have been investigated and the interaction with other language features has been determined. Given the KSOS project's requirement for module generators the implementation team will try to implement them as soon as possible.

6. Structured array constants (i.e. for initialization tables) will be implemented as soon as possible. The EUCLID translator also needs this feature. Structured record constants can be deferred for the time being.

7. Non-standard zones can be deferred for the time being as the initial implementation will allow users to provide their own runtime storage management routines for use with the standard zone.

8. Forward declaration of routines will be implemented.

9. EUCLID compiler will provide Pascal-like type checking and error messages immediately. Checking of imports and exports lists and generation of legality assertions in EUCLID source will be deferred.

10. Manifest expressions will be allowed for labels in Pascal-like case statements. Case labels in variant record definitions and discriminating case statements will be restricted to (named) literal constants.

Points 11-20 in Berson's original message are not of immediate concern to the KSOS project. However, eventual provision of many of the features they address is necessary for KSOS' long range goals of performance and verifiability. (See E, below).

OTHER POINTS

A. It was agreed that instead of delivering a Middle EUCLID translator at the end of July 1978 it would be more beneficial to both the Implementation project and the KSOS project to deliver a translator with more features (especially those noted above) as soon as possible, no later than mid Oct 1978.

B. The implementation team will keep the KSOS project informed of its progress.

C. The implementation team will arrange for the KSOS project to have access to preliminary versions of the EUCLID translator whenever a reasonably stable version of the translator exists. This may involve shipping a tape containing a binary image of the compiler to KSOS or arranging for KSOS to have access to the PDP-11/45 at Toronto.

D. After the telephone conference M. Pliner and D. Bonyun arranged that Bonyun and David Crowe from the implementation team would spend two days at FACC in early August discussing programming in EUCLID with the KSOS project.

E. KSOS felt that the October version of the EUCLID translator would provide a reasonable initial tool for KSOS development. They felt that there was not a great need for a full EUCLID language implementation although some of the deferred compiler features will eventually be essential.

F. The telephone conference resulted in a much better understanding between the two parties. The implementation team has a better idea of the KSOS priorities and schedule. The implementation team understands the KSOS projects concern that the EUCLID translator be on time and correct. The KSOS project is aware of the implementation teams concern to produce a stable, useful tool.

G. After the telephone conference the implementation team made the following assessment of the work to be done after mid-October and the order in which it could be done.

1. Enhancement/maintenance of the mid-October translator as determined by the needs of the KSOS project.

2. Implementation of the translator phase that checks imports and exports lists and enforces EUCLID's access rules.

3. Implementation of non-standard zones.

4. Implementation of INLINE procedures and functions.

5. Implementation of structured record constants.

6. Generation of legality assertions in EUCLID source.

7. Implementation of functions returning non-scalar, non-manifest values.

8. Implementation of other language features mentioned in Berson's points 11-20.

APPENDIX E

ARPANET Message Index

1	1 JAN	To: WORTMAN	78-1, Re: Clara Req 41 (208)
2	3 JAN	Horning at PARC	Re: 78-1, Re: Clara Req 41 (208)
3	2 JAN	To: WORTMAN	78-2, Clara on components of literals, expressions
4	3 JAN	Horning at PARC	Re: 78-2, Clara on components of literals, expressions
5	2 JAN	To: WORTMAN	78-3, Discussion topics for 6-7 Jan Meeting
6	2 JAN	To: WORTMAN	78-4, More Discussion Topics
7	3 JAN	To: WORTMAN	78-5, Cleanup standard functions
8	3 JAN	Horning at PARC	Re: 78-5, Cleanup standard functions
9	5 JAN	To: WORTMAN	78-6, LP#4 Brief Statement of "Sameness" Rule
10	18 JAN	To: mitchell at PARC	78-7, Re Updated EUCLID Report
11	18 JAN	LAMPSON at PARC-MAXC	Re: 78-7, Re Updated EUCLID Report
12	18 JAN	Horning at PARC	Re: 78-7, Re Updated EUCLID Report
13	18 JAN	Horning at PARC	Re: Re: 78-7, Re Updated EUCLID Report
14	18 JAN	To: Horning at PARC	Re: 78-7, Re Updated EUCLID Report
15	18 JAN	To: Mitchell at PARC	78-8, Re 6-7 Jan Meeting Minutes and Mss Code
16	18 Jan	MITCHELL at PARC-MAXC	Re: 78-8, Re 6-7 Jan Meeting Minutes and Mss Code
17	18 JAN	To: WORTMAN	78-9, Re Syntax Revisions
18	18 JAN	Horning at PARC	Re: 78-9, Re Syntax Revisions
19	18 JAN	MITCHELL at PARC-MAXC	Re: 78-9, Re Syntax Revisions
20	18 JAN	LONDON at USC-ISIB	Re: 78-9, Re Syntax Revisions
21	20 JAN	To: WORTMAN	78-10 Re Draft of Jan meeting minutes
22	20 JAN	To: WORTMAN	78-11, Reply to JM Re Syntax Change #19
23	21 FEB	Horning at PARC-MAXC	Re: Re Type Sameness Rule
24	9 MAR	Lampson at PARC-MAXC	Re: Availability of Revised EUCLID Report
25	9 MAR	Horning at PARC-MAXC	Re: Availability of Revised EUCLID Report

26	10 MAR	To: WORTMAN	78-12, EUCLID Problem with WITH and Variant Record Fields
27	10 MAR	Horning at PARC-MAXC	Re: 78-12, EUCLID Problem with WITH and Variant Record Fields
28	10 MAR	To: WORTMAN	78-13, EUCLID Problem with exports and routines.
29	10 MAR	Horning at PARC-MAXC	Re: 78-13, EUCLID Problem with exports and routines.
30	10 MAR	To: WORTMAN	78-14, EUCLID Problem with THUS lists
31	10 MAR	Horning at PARC-MAXC	Re: 78-14, EUCLID Problem with THUS lists
32	10 MAR	To: WORTMAN	78-15, EUCLID Problem 7ith itsTa
33	14 MAR	To: WORTMAN	78-16, Postscript on Type Sameness Rule
34	14 MAR	Lampson at PARC-MAXC	Re: 78-16, Postscript on Type Sameness Rule
35	15 MAR	To: WORTMAN	78-17, Legality Assertions for Parameterized Module Types
36	16 MAR	Lampson at PARC-MAXC	Re: 78-17, Legality Assertions for Parameterized Module Types
37	20 JAN	MITCHELL at PARC-MAXC	Minutes of the Jan 6-7 meeting
38	15 FEB	To: MITCHELL at PARC	Re: Minutes of the Jan 6-7 meeting
39	12 APR	To: Horning at PARC,	Revised EUCLID Report Schedule
40	21 APR	WALKER at USC-ISI	Re: EUCLID verification
41	26 APR	To: Horning at PARC	Revised EUCLID Report
42	26 APR	Horning at PARC-MAXC	Re: Revised EUCLID Report
43	3 MAY	To: Horning at PARC,	EUCLID Type Sameness Rule
44	3 MAY	Horning at PARC-MAXC	Re: EUCLID Type Sameness Rule
45	6 MAY	Horning at PARC-MAXC	Re: Revised EUCLID Report
46	4 MAY	To: WORTMAN	78-18, Type Sameness vs. Storage Allocation
47	4 MAY	Horning at PARC-MAXC	Re: 78-18, Type Sameness vs. Storage Allocation
48	4 MAY	Mitchell at PARC-MAXC	Re: 78-18, Type Sameness vs. Storage Allocation
49	5 MAY	To: Mitchell at PARC	Re: 78-18, Type Sameness vs. Storage Allocation
50	5 MAY	Mitchell at PARC-MAXC	Re: 78-18, Type Sameness vs. Storage Allocation

51	5 MAY	To: Mitchell at PARC	Re: 78-18, Type Sameness vs. Storage Allocation
52	12 MAY	To: WORTMAN	EUCLID Implementation Team Tasks, June-Oct 1978
53	19 MAY	To: Neumann at SRI-U	78-19, Exception Handling in EUCLID
54	22 MAY	Horning at PARC-MAXC	Re: 78-19, Exception Handling in EUCLID
55	22 MAY	LONDON at USC-ISIB	Re: 78-19, Exception Handling in EUCLID
56	25 MAY	Feiertag at SRI-KL	Re: 78-19, Exception Handling in EUCLID
57	25 MAY	Horning at PARC-MAXC	Re: 78-19, Exception Handling in EUCLID
58	23 MAY	To: Bonyun at UCLA-S	78-20, Middle EUCLID
59	23 MAY	GUTTAG at USC-ISIB	middle EUCLID
60	23 MAY	LONDON at USC-ISIB	Re: The Following Message
61	23 MAY	To: LONDON at USC-IS	Re: The Following Message
62	23 MAY	Horning at PARC-MAXC	Re: 78-20, Middle EUCLID
63	24 MAY	Lampson at PARC-MAXC	Re: 78-20, Middle EUCLID
64	24 MAY	Feiertag at SRI-KL	Re: 78-20, Holt/Feiertag Dialogue
65	29 MAY	To: WORTMAN	Re: 78-20, Holt/Feiertag Dialogue
66	30 MAY	Mitchell at PARC-MAXC	Re: 78-20, Holt/Feiertag Dialogue
67	30 MAY	Mitchell at PARC-MAXC	Re: 78-20, Holt/Feiertag Dialogue
68	30 MAY	Horning at PARC-MAXC	Re: 78-20, Holt/Feiertag Dialogue
69	19 MAY	To: WORTMAN, Neuman	The Following Message
70	23 MAY	Horning at PARC-MAXC	New type sameness rule (Part 1 of 2)
71	23 MAY	Horning at PARC-MAXC	New type sameness rule (Part 2 of 2)
72	30 MAY	Mitchell at PARC-MAXC	The revised report
73	1 JUN	To: Mitchell at PARC	Re: Revised EUCLID Report
74	1 JUN	To: Mitchell at PARC	Re: Nonscalar Functions
75	5 JUN	LONDON at USC-ISIB	EUCLID mades the ads
76	14 JUN	Lampson at PARC-MAXC	78-13, 14, 15, 16, 17
77	19 JUN	LONDON at USC-ISIB	Proof Rule Galley Proofs
78	20 JUN	LONDON at USC-ISIB	Re: Proof Rule Galley Proofs
79	20 JUN	Lampson at PARC-MAXC	Re: Proof Rule Galley Proofs
80	21 JUN	To: Lampson at PARC	Re: Proof Rule Galley Proofs

81	21 JUN	HORNING at PARC-MAXC	Re: Proof Rule Galley Proofs
82	21 JUN	LONDON at USC-ISIB	Re: Proof Rule Galley Proofs
83	21 JUN	Mitchell at PARC-MAXC	Re: Proof Rule Galley Proofs
84	21 JUN	Mitchell at PARC-MAXC	Re: Proof Rule Galley Proofs
85	23 JUN	Horning at PARC-MAXC	EUCLID Report Delay
86	25 JUN	Neumann at SRI-KL	Re: EUCLID Report Delay
87	25 JUN	Horning at PARC-MAXC	Re: EUCLID Report Delay
88	26 JUN	To: WORTMAN	78-21, Revised Report vs. the Coarse-toothed comb
89	26 JUN	To: LONDON at ISIB	Re: 78-21, Revised Report vs. the Coarse-toothed comb
90	26 JUN	To: mitchell at PARC	Zones in EUCLID
91	26 JUN	Mitchell at PARC-MAXC	Re: Zones in EUCLID
92	27 JUN	To: Lampson at PARC	Editing of Revised Report
93	27 JUN	Lampson at PARC-MAXC	Re: Editing of Revised Report
94	28 JUN	To: WORTMAN	Re: Revised Report
95	29 JUN	To: WORTMAN	EUCLID at ACM '78
96	30 JUN	Withington,MDruid	EUCLID WP #54
97	30 JUN	To: Withington,MDruid	Re: EUCLID WP #54

Distribution List for Technical Reports

ARPA, Attn: Program Management — 2 copies
1400 Wilson Blvd.
ARLINGTON, VA 22209

Dr. G.X. Amey — 1 copy
CRAD DST(SE)4
Department of National Defence
101 Colonel By Drive
OTTAWA, Canada K1A 0K2

Defense Documentation Center (DDC) — 12 copies
Cameron Station
ALEXANDRIA, VA 22314

Letters of Transmittal sent to:

Mr. Ken Laver
Science Procurement Branch, CCC
11C1 Place du Portage, Phase III
11 Laurier Street
HULL, Quebec K1A 0S5

Defense Contract Administration Service
Management Area, Ottawa
219 Laurier Avenue, West
6th Floor
OTTAWA, Canada K1A 0S5

Mr. Steve Walker
OSD-CCCI
The Pentagon
WASHINGTON, D.C. 20310